CMHC SCHL

Helping to house Canadians

Canada

USING EXPERT SYSTEMS TO CHECK COMPLIANCE WITH MUNICIPAL BUILDING CODES

By Eric J. Heikkila
and
Edwin J. Blewett

April, 1991

CMHC Project Manager: Wayne Webster

This project was carried out with the assistance of a grant from Canada Mortgage and Housing Corporation under the terms of the External Research Program (CMHC CR File 6585/H59). The views expressed are those of the authors and do not represent the official views of the Corporation.

92-04385/

Acknowledgements

The authors gratefully acknowledge support from the Canada Mortgage and Housing Corporation's External Research Program. We are also indebted to City of Vancouver staff who participated in this pilot project and to three anonymous referees who provided helpful comments on an earlier draft. The usual caveat applies.

Dr. Eric J. Heikkila is assistant professor of urban and regional planning at the University of Southern California. He has published extensively on issues arising from urban land use regulation, with particular emphasis on the impacts of changing land use. The research reported on here is part of an ongoing research program into the role of technological innovation in land use planning.

Dr. Edwin J. Blewett is principal of EB Economics, a private consulting firm specializing in decision support and information system design. His more recent assignments have focused increasingly on the use of computer graphics and intelligent information systems as components of an integrated approach to decision making in complex environments.

Abstract

Using Expert Systems to Check Compliance with Municipal Building Codes

by Eric J. Heikkila and Edwin J. Blewett

The regulation and enforcement of building codes is a primary function of local governments. Although the regulations are typically numerous, complex and highly interactive, the plan checkers who administer them are often relatively inexperienced. A prototype based on the Building and Zoning & Development ByLaws governing spatial separation between single family residential dwellings in Vancouver, Canada demonstrates that expert systems can be a useful aid to plan checkers in striving for consistent bylaw enforcement.

SUMMARY

A microcomputer based expert system prototype to assist the Vancouver Building By-law department check spatial separation between structures was developed.

The work involved identifying the relationships affecting the fire separation distances between dwellings. The report defines the various definitions and relationships used for reviewing allowable configurations of openings. As well, technical considerations related to the selection of expert system shells (reasoning mechanisms) and interface specifications (graphical vs dialogue) alternatives etc are noted.

The benefits of such a product are consistency, speed, and accuracy in determining plans compliance with a set of municipal bylaws. As well, there is the potential that bylaw and code changes may be readily incorporated into the plan checking process and that bylaw inconsistencies can be identified.

Trial use of the prototype system suggested that it was useful training for the plan checkers, and that enhancing the graphic capabilities of the system's user interface would further increase its benefit to users. Further, there appears to be considerable potential to expand the use of knowledge based (expert) systems to other areas.

Utilisation de systèmes experts pour vérifier la conformité aux codes de bâtiment municipaux

RÉSUMÉ

On a mis au point un prototype de système expert piloté par micro-ordinateur pour aider la ville de Vancouver à contrôler la conformité de l'isolement des bâtiments aux règlements municipaux.

Il a fallu déterminer les éléments influant sur la distance des séparations coupe-feu entre les habitations. La rapport fait état des divers définitions et éléments qui ont servi à examiner les différents types d'ouvertures acceptables. De plus, on relate les considérations techniques qui ont conduit au choix des systèmes experts génériques (mécanismes de raisonnement) et des spécifications d'interface (graphique ou de dialogue).

Un tel produit permet de déterminer avec constance, rapidité et précision si les plans d'un ouvrage respectent les règlements municipaux. En outre, il offre la possibilité d'intégrer facilement au processus de vérification les changements apportés aux codes et aux règlements et de repérer les anomalies dans les règlements municipaux.

L'essai du prototype a mis en évidence son utilité pour la formation des vérificateurs de plan. De plus, si l'on pouvait améliorer les capacités de l'interface graphique homme/machine, les utilisateurs y gagneraient beaucoup. Enfin, il semble exister un potentiel considérable pour étendre l'utilisation des systèmes experts à d'autres secteurs d'activités.

CMHC SCHL

Helping to house Canadians

Question habitation, comptez sur nous

National Office

700 Montreal Road
Ottawa, Ontario
K1A 0P7

Bureau National

700 chemin Montreal
Ottawa (Ontario)
K1A 0P7

Puisqu'on prévoit une demande restreinte pour ce document de recherche, seul le sommaire a été traduit.

La SCHL fera traduire le document si la demande le justifie.

Pour nous aider à déterminer si la demande justifie que ce rapport soit traduit en français, veuillez remplir la partie ci-dessous et la retourner à l'adresse suivante :

Le Centre canadien de documentation sur l'habitation
La Société canadienne d'hypothèques et de logement
700, chemin de Montréal, bureau C1-200
Ottawa (Ontario)
K1A 0P7

TITRE DU RAPPORT: _____

_____

Je préférerais avoir ce rapport en français.

NOM _____

ADRESSE _____
rue *app.*

---
ville *province* *code postal*

No de téléphone ( ) _____

1946 CMHC SCHL 1991

TEL: (613) 748-2000

Canada Mortgage and Housing Corporation Societe canadienne d'hypotheques et de logement

Canada

Using Expert Systems to Check Compliance with Municipal Building Codes

INTRODUCTION

Expert systems are a branch of artificial intelligence designed to replicate human expertise over a general set of problem domains. Expert systems may be thought of as logic-based rule processing software, where the rules embody the expertise in question. Early applications included medical diagnoses, storage of hazardous chemicals and military deployment. Applications to the field of planning arose somewhat later, with Ortolano and Perman (1987), Han and Kim (1989) Kim, Wiggins and Wright (1989) and Heikkila (1990) being early attempts to assess expert systems' general potential in planning.

An expert system consists of three basic components, a knowledge base, inference engine and user interface. What makes expert systems distinct from conventional programing is that the inference engine (the "thinking machine") can be disembodied from the rules which it processes (Ortolano and Perman, 1987). This permits the development of expert system shells containing the inference engine and user interface, and indeed commercial development of these shells is a flourishing subindustry. Given an expert system shell, the task of the expert system developer is to translate expert knowledge for the problem domain in question into precise rules that conform to the logical and syntactical requirements of the shell.

Han and Kim (1989) recently posed the question, "Can expert systems help with planning?". The planning functions they review are primarily complex decision environments with no single best or correct answer. Examples of multidimensional planning problems addressed by Han and Kim include landfill site selection, urban design, and bus route choice. These are decision contexts that clearly do *not* satisfy the criteria for expert system applications set out by Waterman (1986), Silverman (1987) and Goodall (1985), who advocate crisply defined problems with certain solutions and domain rules. As Han and Kim point out, these guidelines are seldom likely to be met in a planning context. Nonetheless, they conclude that expert systems can work well in such contexts, particularly when combined with management database systems, geographic information systems or decision support systems.

Not all planning functions are so nebulously defined, especially those pertaining to routine administrative or regulatory practices. Administration of local government building or zoning bylaws is a case in point. These bylaws are the most concrete articulation of the planners' vision, and they are the most immediate focus of the continuing dialogue between city hall and the development community. While the process by which these bylaws evolve is complex and at times obscure, the bylaws themselves are a set of codified rules that govern the development process explicitly.

In residential areas municipal bylaws typically focus on a range of concerns, including yard setbacks and building standards. This is a relatively high volume area for building permits and is handled by a cadre of front line staff whose job it is to check whether submitted plans for residential building renovations or construction conform to the written code of the building bylaws. This plan-checking function is often a difficult one to perform well because the bylaws themselves are often highly complex, confusing and mutually interactive. Despite the complexity and high public profile of their jobs, plan checkers are typically quite low in seniority, receive relatively low wages and are the first to suffer the frustration exhibited by developers or homeowners whose building permits are denied. Not surprisingly, staff turnover among plan checkers is quite high, further perpetuating this vicious cycle of inexperience and frustration.

This paper assesses the feasibility of using expert systems as a plan checking tool. To this end we have developed and tested a prototype expert system based on the fire separation component of the City of Vancouver Building By-Laws. Fire separation in this context refers to the relationship between the distance of a dwelling unit from the neighboring property and the percentage of the dwelling's exposing building face that can be devoted to windows, doors and other unprotected openings. In the following two sections we describe the Vancouver bylaws that govern fire separation and how plan checkers determine whether applications comply with these bylaws. This is followed by a general discussion of why expert systems might be thought to be an appropriate technology in this context. Subsequent to that is a description of the specific prototype expert system we developed for this purpose and an assessment by plan checking staff of its performance. We also speculate on the potential for more widespread applications of this type. A number of general issues emerge in this discussion, both technical and organizational. On balance, however, we are persuaded that the benefits of consistency, speed and accuracy afforded by expert systems makes them a logical tool for checking compliance of submitted plans with a set of municipal bylaws.

FIRE SEPARATION AND THE VANCOUVER BUILDING BYLAWS

The motivation for fire separation bylaws is to reduce the risk of fire spreading from one dwelling to another. While the details of the bylaws and of their application in practice can be quite complex, the basic principle is straightforward: more unprotected openings closer to an adjacent dwelling increase the fire threat. If the **exposing building face** is a single plane, then the percentage of unprotected openings allowed by the bylaws increases monotonically as the distance of the exposing building face is further away from the property line. The precise relationship between this **minimum limiting distance** and the **allowable unprotected openings percentage** (AUOP) depends upon other considerations as well. For example, if either of the dwellings (the one being considered or the one adjacent) have built in sprinklering systems, or if the dwelling in question has protective fire-resistant glazing on all openings, then an enlarged AUOP may be warranted. The total square footage of openings permitted on the exposing building face is obtained by multiplying the total area of the exposing building face by the AUOP for the minimum limiting distance.

Further complications may ensue as the exposing building face is broken up into planes of varying distances to the neighboring property line. In this case a plan checker first examines whether the proposed openings exceed the AUOP prescribed for the closest plane. If not, the fire separation requirements are clearly satisfied, since the AUOP increases with distance from the property line. However, if this preliminary test fails, then the plan checker proceeds with a more elaborate test to determine whether the proposed openings are allowed by the building bylaws that govern fire separation. The principle underlying this more elaborate test is as follows. Each opening on the exposing building face has associated with it a certain **limiting distance**, which may be conveniently thought of as its *effective* distance from the property line, from the perspective of fire threat. The limiting distance for any opening depends upon its position relative to flanking walls and relative to openings on those flanking walls. The strengths of an expert system are best displayed in the search for the correct limiting distance, because this is where the structure of plan checking rules and procedures becomes most complex.

Once the limiting distance has been determined for any opening, a procedure known as **ratioing** is undertaken. The purpose of this exercise is to calculate a **reduction factor** for each opening that is used to effectively reduce the actual opening area to some smaller number, where larger limiting distances will be associated with smaller effective openings (i.e., with smaller reduction factors). Of special interest is the minimum limiting distance over all openings on the exposing building face. For each limiting distance, including the minimum, there is a unique AUOP (subject to additional information on sprinklering and protective glazing). The reduction factor for any opening, which is never greater than one, is defined as the ratio of the AUOPs for the minimum limiting distance and the limiting distance, respectively. That is,

$$(1) \qquad RF_j = AUOP^* / AUOP_j$$

where RF_j is the reduction factor for the j^{th} opening, $AUOP_j$ is the allowable unprotected opening percentage for the j^{th} opening, and $AUOP^*$ is the AUOP for the minimum limiting distance (and so $AUOP^* \leq AUOP_j$). The "reduced area" for each opening (ROA_j) is calculated as the product of the actual opening area (AOA_j) and the reduction factor. That is,

$$(2) \qquad ROAj = AOAj * RFj$$

The plan checker will approve the proposed plan for fire separation if the total reduced area for all openings on the exposing building face is within the allowable total for the minimum limiting distance.

Thus, to recapitulate, the fire separation portion of Vancouver's building bylaws are concerned with the risk of fire spreading between adjacent dwelling units. As the distance from the dwelling to the property line increases, so too does the percentage of the exposing building that can be allotted to doors, windows and other unprotected openings. In determining whether a submitted plan conforms to the fire separation bylaws, the plan checker looks first to see whether the actual openings fall within the

prescribed maximum. If so, then the fire separation bylaws are satisfied. If not, and if the exposing building face resides on multiple planes of varying distances from the property line, ratioing is used to reduce the effective areas of openings whose limiting distances exceed the minimum limiting distance. Much of the plan checker's attention is directed towards the search for each opening's limiting distance. The next section provides an overview of how one determines what the limiting distance is for any one opening.

DETERMINING THE LIMITING DISTANCES FOR UNPROTECTED OPENINGS

The limiting distance for an unprotected opening is the effective distance of that opening to the property line from the perspective of fire threat. Vancouver's fire separation bylaws are based on thermodynamic experiments conducted decades ago to assess how various building configurations affect heat and flame radiation. Several crucial factors are embedded in the bylaws, including (i) distance of an unprotected opening from any flanking walls, (ii) the existence of unprotected openings on flanking walls, and (iii) the length of any flanking walls. For the purposes of this study, it is irrelevent whether the fire separation bylaws in place in Vancouver are optimal, thermodynamically or otherwise. Instead, what is important is to ascertain whether current practice for checking conformity with the fire separation bylaws can be replicated via expert systems.

Current practice is best described with reference to figure 1, which shows an opening W on the exposing building face of a courtyard with two flanking walls, where each flanking wall also has an opening. In this example there are five candidates for the limiting distance of opening W. One candidate is simply the actual distance of the opening to the property line. Another candidate for limiting distance is termed "projection left" (PL), the distance of the edge of the left flanking wall from the property line plus the distance of opening W to the left flanking wall. When an opening is adjacent to a flanking wall, as is the case here, the wall has the effect of concentrating the heat and flame, thereby increasing the distance of radiation. The closer an opening is to the flanking wall, therefore, the further forward is its effective distance projected, and so PL becomes smaller. A third candidate for the limiting

distance of opening W is "flanking opening distance left" (FODL), defined as the distance to the property line of the edge of the opening in the left flanking wall that is nearest to the property line. If opening W is too close to the left flanking wall, then there is a danger that fire will spread to that wall and that the flanking opening will itself become a source of flame and heat radiation, and so FODL becomes a candidate limiting distance. Two additional candidate limiting distances are PR and FODR, based on the same concepts applied to the right flanking wall. In general, the candidate that yields the smallest limiting distance is the effective limiting distance.

Several exceptions apply. First, if a line drawn outward at a 45 degree angle from opening W does not intersect the left flanking wall, then neither FODL nor PL will be the limiting distance. The reasoning behind this rule is that flame radiates from an opening at a 45 degree angle, and so if the conditions described above are met, the flanking wall is not in danger from this opening. Another consequence of the 45 degree angle radiation is the "5 foot 8inch rule", which states that neither FODL nor PL will be the limiting distance if the opening is more than 5 feet 8 inches from the left flanking wall. If the opening is just 5 feet 8 inches from the flanking wall, then a 45 degree line drawn to the flanking wall from the opening will be 8 feet in length, and the effect of heat and flame radiation on flanking walls is considered negligible beyond 8 feet. Similar rules apply to the right flanking wall. In the example depicted in figure 1 the limiting distance is 6 feet, corresponding to FODR.

As the above illustration and examples show, the plan checkers' job can be tedious and exacting. The basic courtyard configuration described above has 57 distinct cases, and careful reasoning is required to determine which case applies in any one instance. Figure 2 provides a rule from the VP Expert prototype knowledge base corresponding to one of these 57 cases. The number of cases may in turn be multiplied several fold to allow for various sprinklering or protective glazing options. Additional sets of cases arise where there is only one flanking wall, right or left.

Moreover, spatial separation is not the only item that plan checkers look for. They must also determine whether each submitted plan complies with the bylaw requirements for exiting and for yard setbacks. Thus, it is not surprising to find that staff turnover among plan checkers is high. In the case of Vancouver, the senior-most plan checker had fourteen months experience, and eighty percent of the plan checking staff had been on the job less than six months.

A PROTOTYPE EXPERT SYSTEM FOR CHECKING COMPLIANCE WITH VANCOUVER'S FIRE SEPARATION BY-LAWS

For this research we undertook to build a prototype expert system to check compliance with Vancouver's fire separation bylaws as they apply to single family residential properties within the city (those zoned RS-1). The prototype was developed over a three month period using the software shell VP Expert and contains approximately one hundred fifty rules drawn primarily from sections 9.10.14 of the City of Vancouver *Building Bylaw No. 6134* and from the Permit and Licence Department's *Fire Protection Manual*. No attempt was made to link the prototype to the City's other data bases, although there is clearly potential for doing so. Design emphasis for the prototype was primarily on logical processing of rules and only secondarily on user interface. In particular, the user interface relies on a simple query mode and does not have any graphics capabilities. The prototype examines fire separation only, and does not address exiting requirements or yard setbacks. City of Vancouver staff from the Department of Permits and Licenses and from the Division of Computer Services were actively involved in the project, but the prototype itself was built by the authors alone. The purpose of the prototype was to test whether expert systems can indeed replicate human expertise in checking whether submitted plans conform to the fire separation bylaws.

There are a host of reasons for believing a priori that the plan checking function is amenable to expert systems application. First, precedents elsewhere are encouraging. Hosking et al (1987) in New Zealand and Leary (1989) in England undertook early attempts to encode local government building or development bylaws, and both found that the availability of a written code significantly reduces the effort required for development. While there are instances of researchers reporting that a formal written code was sufficient resource material for the knowledge acquisition phase (Sergot et al, 1986), it is more often the case that human experts are called upon as an additional resource for interpretation of vague or incomplete bylaws. For the application reported upon here, the written bylaws and supplementary interpretive documents were indeed useful, but extensive and ongoing consultations with plan checking staff were essential.

One clear advantage of the plan checking function over other planning applications is that the relevent bylaws, while sometimes confusingly worded, are nonetheless definitive.

That is, if the premises of the bylaw are met, then the conclusion is valid with complete certainty. This avoids reliance on subjective "confidence factors" or other methods for dealing with decision making under uncertainty.
Additionally, the bylaws do provide a tangible focus for the transcription of facts and rules for the knowledge base.

Partial automation of the plan checking function suggests other important potential benefits based on increased staff productivity. Accuracy, consistency, speed and training are all likely to be enhanced with the aid of expert systems. On several occasions during the knowledge acquisition phase we encountered instances where plan checking staff came to differing conclusions regarding an application. On other occasions a single plan checker would reach different conclusions on similar applications. Reduction or elimination of inconsistency between individuals or over time is one of the key potential benefits of a more automated plan checking operation.

Automation of the plan checking process also suggests the possibility of linkages to other data bases. Most local governments maintain large data bases on real properties within their jurisdiction. These land use bases may be part of a geographic information system, a tax billing system or general utility land use information system. Data drawn from these data bases may supplement the expert system consultation. For example, calculation of front and back yard setbacks may require data on adjacent parcels. Similarly, information on building footprints obtained via an expert system plan checking consultation may be used to gradually add detail to the agency's larger parcel files. Thus, there is reason to think that expert systems could eventually become a more integral component of local government information systems. Although the City of Vancouver maintains a permit tracking system (PALS) on its mainframe, the prototype we developed was explicitly designed as a stand-alone demonstration model. The City may assess the desirability of a link to PALS at a later date.

An additional potential benefit from a partially automated approach to plan checking is enhanced documentation. Currently the calculations undertaken by plan checkers are scratched onto the plan itself in a highly unstructured format. If the builder should choose to appeal a rejection at a later date the plan checking process is usually repeated in full for want of documentation. In contrast, most expert system shells feature detailed explanation capabilities. After a consultation is completed, the user can query the system on "HOW" a decision was arrived at, i.e., which rules were fired and in what

order. Additionally, the user can search systematically through a list of variables to determine what their values are at any time during the consultation. Most sophisticated expert systems also contain "WHAT IF" capabilities, allowing the user to ask how the conclusion might vary were certain variable values to differ. This capability is especially useful in a plan checking context as the focus of the discussion shifts from screening to prescription. For example a builder might, upon having his plan rejected, seek to determine whether altering the size or location of a specific opening would result in a favorable decision. The entire consultation, together with related queries can be kept in an ASCII, database or spreadsheet file for further review at a later date. This is especially useful in the plan checking context where staff turnover is habitually a problem.

Expert systems have other advantages over conventional programming from a plan checking perspective. First, bylaws and other written regulations translate rather naturally into the rule syntax of expert systems. Moreover, one can rely on the expert system shell itself to undertake the logical processing of rules, thereby permitting one to focus on the construction of the rules themselves and on various aspects of the user interface. Perhaps most importantly, however, expert systems are goal-oriented rather than process-oriented, and this is a decided advantage for plan checking applications. A conventional programming approach seeks to mimic step by step the process by which a decision is arrived at. In expert systems, in contrast, the logical sequence is endogenous. The expert system is searching for an answer to a specific question ("Does this building permit application satisfy the fire separation requirements of the building bylaw?"). The chain of reasoning it follows to determine a final answer depends entirely upon the intermediate answers it receives during the course of the consultation. For example, an entire set of questions may (or may not) be set off in response to a finding that the building is not sprinklered. Most expert systems can provide a "trace" of the reasoning process used to arrive at conclusions. Figure 3 provides a small exerpt from a trace for a trial consultation. It is this endogenous reasoning capability that qualifies expert systems as a branch of artificial intelligence.

IN SITU PERFORMANCE EVALUATION OF PROTOTYPE

The prototype expert system we developed was put into trial use by the Vancouver Permits & Licenses Department (hereafter referred to as the Department). Plan checkers used the prototype to test a wide range of single family permit applications over a three month period and prepared a summary of their findings in the form of an internal memorandum (City of Vancouver, 1991). As developers of the expert system we were not part of the evaluation team, but we were invited to a special meeting of the plan checkers to learn of and comment on their findings. This section summarizes the Department's evaluation of the prototype as presented at that meeting and described in their internal memorandum. Stated simply, the Department concluded that the expert system prototype performed remarkably well with respect to consistency and accuracy, but that the data entry or user interface mechanism was somewhat cumbersome. All discussants agreed that a graphics oriented user interface would be a vast improvement over the prototype's simple query mode.

One problem identified by the plan checkers stems from the immediate hardware environment. Twelve plan checkers were compelled to share the single personal computer that housed the expert system software. This computer was kept on a small table in a remote corner of the plan checkers' workspace, and was not set up with adequate table space for laying out the building plans comfortably. In contrast, each plan checker has a computer terminal on his/her desk that is linked to a permit tracking system on the City's mainframe. This hardware constraint presented a significant hurdle to initial use of the expert system. Had it not been for the Department's determination to test the system it may not have been used at all. However, after gaining exposure to the expert system through the testing procedure the plan checkers did find that walking from their desks to the prototype system was worthwhile, but only for more complex plans. This experience reinforces our view that an expert system's acceptance and use depends in no small measure on ergonomic or institutional considerations, in addition to the detailed design of the software itself.

Each of the plan checkers interviewed vouched for the accuracy of the expert system. In all cases where the manual calculations deviated from the expert system, further investigation led the plan checkers to conclude that the prototype's calculations were the correct ones. This finding alleviated a concern we had earlier that the discovery of some mis-specifications on our part of the knowledge base rules might prejudice the plan

checkers against expertise systems technology. More importantly, from the Department's perspective, the accuracy and consistency of the expert plan checking system was seen as a potential means of reducing the City's risk of liability claims stemming from incorrect or incomplete plan checking. Annual insurance premiums for errors and omissions to cover this potential liability for the City recently exceeded one million dollars Canadian, so the value of enhanced accuracy is a significant factor in the City's reckoning.

The Department concluded that its accuracy and consistency would also make the expert plan checking system a useful training tool, although the prototype was not specifically designed with this function in mind. This is seen to be particularly valuable in light of the Department's astonishing staff turnover rates. Although the Department maintains a staff of just ten plan checkers, it lost twenty-two over the previous twelve months due to resignations, terminations, or promotions. Consequently, there is a steady demand for staff training so that continuity and consistency can be maintained from year to year. Moreover, the bylaws themselves are subject to frequent change, and this generates an additional demand for staff training. An example of this occurred during the course of our project, where an entire new set of regulations was introduced governing which trees are subject to protection under the bylaws.

No consideration was given in the Department's findings to potential linkages with other databases maintained by the City. It appears that the expert system is sufficiently novel that staff were quite preoccupied with more immediate concerns about how to use it and whether or not it was reliable. This attitude on the part of the Department may also reflect the terms of reference for the initial prototype, which was clearly intended as a stand-alone demonstration module.

One of the most encouraging results reported by the Department concerned the time savings associated with more complex permit applications, defined by the Department as those which require ratioing and which have flanking walls that need to be considered. According to the Department these account for twenty-five to thirty percent of the hundred and fifty to two hundred residential building permit applications received in a typical month. Processed manually by an experienced plan checker, a complex permit application requires thirty to forty-five minutes to review. In contrast, the same plan checker was reviewing complex applications in ten to fifteen minutes after becoming familiar with the sequence of questions. Based on the more conservative figures for

spatial separation alone, this translates into a savings of over fifteen and one-half person-hours per month.

For simple applications the results were less clearcut. On the one hand the plan checkers were impressed with the speed with which the prototype calculated the allowable unprotected openings percentage for any given limiting distance. But this admiration was tempered by the elaborate set of questions that seemed better suited to more complex applications. While the judgements made by the prototype were accurate and consistent for the simple applications, most plan checkers found the time savings offered by the prototype did not justify leaving their desks in order to engage in an expert system consultation. This is representative of the kinds of design issues that prototypes can uncover.

A related criticism staff had of the prototype concerns the lack of a graphics interface. Plan checkers work directly with information obtained from the submitted building plans. These plans show the intended location of all doors, windows and flanking walls and provide the necessary detail on elevations and limiting distances. Much of this information is conveyed visually yet the prototype relied on a query mode. Thus, as illustrated in figure 4, the plan checker was required to translate the visual data into a textual format, while a well designed graphics interface would obviate that requirement and could considerably shorten the time needed for data entry. Based on the discussion above it is reasonable to expect that a graphics module together with a relaxation of hardware constraints could readily shave an average of five minutes from each simple plan submitted. Using the same conservative figures on volumes of permit applications described earlier this would translate into an additional savings of nine person-hours per month. One might reasonably expect that these savings would be enhanced if the expert system's domain were expanded to include other aspects of the plan checkers function such as yard setbacks or exiting requirements.

To recapitulate, the Department's experience with the prototype was highly favorable, particularly with respect to the system's accuracy and reliability. Staff viewed the prototype as a useful means of gaining familiarity with the potential of expert systems applications to plan checking, and in focusing discussion at an early stage on specific design issues. Reducing hardware constraints and enhancing the graphic capabilities of the system's user interface were seen as priority areas for future improvements. Perhaps the most telling assessment of the prototype expert plan checking system is that it is still in use.

FUTURE PROSPECTS FOR EXPERT PLAN CHECKING SYSTEMS

Based on the trial use of this prototype, our overall assessment of expert systems applications to the plan checking function is quite positive. Nonetheless there are a number of issues, both technical and organizational, that should be addressed by planning agencies interested in following this avenue. The technical issues pertain primarily to the features one looks for in an expert system shell while the general issues concern the manner in which the expert system fits into the organizational fabric of the plan checking unit.

Technical Issues

Tello (1988) and Wiggins et al (1989) provide useful reviews of a range of expert system shells. Shells vary in terms of price, reasoning mechanisms, user interface and graphics capabilities, ability to link to external data files, methods for dealing with uncertainty and properties of inheritance. Prices of expert system shells vary from a few hundred to many thousands of dollars. These costs are quite small relative to plan checking budgets and so one presumes that cost would not in itself be a major inhibiting factor. However, as the organization is learning more precisely what its requirements are, there is much to be said for purchasing a relatively inexpensive shell for prototyping purposes so that more seasoned judgement can then be applied to the selection of the shell to be used for full scale implementation.

Forward chaining and backward chaining are the two main types of reasoning processes found in expert systems. In either case the basic rule structure is "If PREMISE, then CONCLUSION". Forward chaining begins with the known premises and attempt to infer conclusions implied by those facts. Backward chaining is more goal oriented, beginning with the desired conclusion and working "backwards" to determine whether the corresponding premises are met. Both chaining methods may work through lengthy chains of rules before finally linking the initial premise with the final conclusion. Pedersen (1989) argues that forward chaining is more appropriate to situations where the objective is to flesh out all relevent facts associated with a given premise, and where the goal is not necessarily known. Backward chaining, in contrast, is more appropriate where the objective is to infer one key fact ("fire separation bylaw requirements are

met"). Forward chaining is typically more useful as an aid in configuration, planning and interpretation while backward chaining is best for classification, selection or diagnosis. For the purpose of plan checking, backward chaining is clearly more appropriate for the limited objective pursued here, which is to determine whether or not a given plan conforms to the bylaw requirements. If one wished to extend the expert plan checking system's scope to include suggested modifications to the plan, then elements of forward chaining might be usefully introduced.

A number of options exist for the user interface mechanism, with considerable potential for graphic interfaces. For this prototype a simple query mode was adhered to. Most expert system shells come equipped with this capacity, and can be readily trained to prompt the plan checker with well phrased questions eliciting information required during the course of a consultation. Another option is to make use of a spreadsheet template for inputting data. The data would then be accessed by the expert system directly from a spreadsheet file for each application. A data template input method has obvious advantages from a documentation perspective and may lead to even quicker processing speeds. One disadvantage of a data template is that much of the reasoning displayed by the system would then be less visible to the applicant and to the plan checker. The user interface can also be supplemented by a more graphic approach. The applicant might first select from a host of generic building types and then could be prompted to fill in information about each opening and each flanking wall in a visual pattern that mirrors what the plan itself contains. As noted above, this kind of graphics interface was identified by the Department as a priority for future development.

Some of the more sophisticated expert system shells feature an inheritance capability whereby objects of interest may "inherit" the characteristics of parent objects. For example an exposing building face that resides in a courtyard may be a special instance of the parent, exposing building face. Then, all rules that apply to exposing building faces in general would also apply to the specific case of an exposing building face found in a courtyard. The expert system shell used to develop the prototype reported upon here does not have an inheritance capability, but this did not appear to be a handicap for prototyping purposes. In contrast, Leary (1989) used software that allowed for one level of inheritance, and reported that it was a useful feature for his application.

Organizational Issues

An expert plan checking system will only be useful if it conforms to and enhances existing practice. A critical factor is acceptance of the system by the plan checking staff, for it is their expertise that is being captured by the system. Any divergence by the system from established routine could lead to its abandonment by skeptical staff. An example of this tension between technical purity and contextual sensitivity arises in the use of backward chaining expert systems. Standard practice is to invoke as little procedural content as possible when using backward chaining. This way, the inference engine itself is unencumbered in its search for the desired conclusion.

In the plan checking context we found that it was not always practical or desirable to avoid procedural programing. In a preliminary version of the prototype, staff would at times become confused by a sequence of questions that differed from the order they were accustomed to. A plan checker might ask, "Why is it asking me about the flanking window before asking me how far this opening is from the property line?". When the expert system is allowed to reason independently, there is in fact little reason to believe that the sequence of questions will correspond to established practice. So long as the rule base is complete and correct, this should not be a problem from the perspective of reaching the desired conclusion. However, it can affect staff's perception of the system and their willingness to use it. We found it useful therefore to delineate two distinct functions of the plan checking system, (i) to obtain pertinent data regarding the configuration of the proposed dwelling unit and then (ii) to subject those facts to the logical processes of the expert system. The former function is more appropriately procedural while the latter is best subject to pure backward chaining reasoning.

Another issue from an organizational context may be summarized by "Do as I do, not as I say". It may happen on occasion that routine practice is based as much on an accepted interpretation of the bylaws as on the bylaws themselves. In such instances it was our decision to base the prototype on existing practice to encourage its use. While this did not prove to be a problem in this particular application, we can envision situations where senior staff may not wish to formalize or document existing practice via the expert system for fear that controversy over interpretation of the bylaws may ensue. More generally, we found several instances where instructive staff discussions were prompted by the need to codify existing practice.

Staff acceptance of an expert plan checking system will also depend vitally upon how they perceive it will affect their own long term job security, where clearly a more favorable response will be engendered if they believe that the system will enhance their productivity and reduce the tedium of their jobs while in no way threatening to replace plan checking staff. For the research reported upon here, both the presence of a strong employee union and rapid turnover in plan checking staff mitigated overt concerns on this point. Moreover, the immediate gains in efficiency were welcomed by staff who are at times overburdened by the volume of plans to be checked.

The presence of a full scale expert plan checking system would have staff implications beyond the front line of plan checkers. As with any automated system there is an ongoing need for maintanence and fine tuning. The bylaws themselves are hardly static, and existing practice evolves slowly but steadily in response to test cases that arise and whose outcomes are assimilated into future interpretations of the bylaws. It is essential that the system be able to evolve in a parallel fashion if it is to remain useful. Some in-house maintanence supplemented by outside expertise is probably in order, with the precise combination depending upon available resources. Similarly, copyright or maintanence considerations may also affect one's choice of software.

Finally, in designing an expert plan checking system one must consider the question, "Who is intended to run whom?". It would be a mistake to try and design a system that does it all. In addition to being practically infeasible, it is not desirable from a human standpoint. An expert system should be staff enhancing rather than staff replacing. To this end the design focus should be on elimination of tedium and the highlighting of the logical path over which decisions are reached. The system can and should be designed to instruct as it infers, so that plan checkers become more rather than less actively involved in the process. Not only does this approach enhance the quality of the plan checkers' work and their job satisfaction, it is also essential for enabling them to advise builders about their options in the event that the plan is rejected.

There is much to be said for first developing a prototype based on a relatively narrow set of bylaws, particularly where an expert system is new to the organization. A prototype, while smaller than a full scale system, can be quite wide ranging in its functionality. Numerous experiments can be undertaken with the coding of rules or with the user interface. As with most endeavors, it is only through active involvement that one comes to appreciate the subtle differences underlying seemingly innocuous

choices about how to proceed. Even if, on the basis of having developed a prototype, one concludes that a different choice of shell is called for or that some other fairly radical changes are needed, there is often much that can be salvaged. For example, the prototype rules may be thought of as an additional source of knowledge to be encoded into the new system or a data input mechanism may still retain its usefulness. Or, as was the case in this research, the prototype can be developed to stand alone so that its usefulness for a specific purpose is not dependent upon future development decisions.

CONCLUSIONS

Based on the prototype developed here it would appear that the plan checking function of local governments is a natural application for expert systems. Plan checking is complex yet relatively unambigous. The relevent bylaws are an excellent initial source of written code for knowledge acquisition, particularly when supplemented by staff training manuals. In addition to the fire separation bylaws, there is clear potential for similar applications to exiting and yard setback requirements.

Notwithstanding the above, there are numerous issues to consider when developing an expert plan checking system. Technical issues include the type of reasoning mechanism to employ, user interface specifications, dialogue versus graphics and inheritance capabilities. Organizational issues include procedural programming versus endogenous reasoning, determining the appropriate scope for an application, and design guidelines to ensure that the system is staff enhancing. These issues complicate the decision to employ an expert plan checking system, and how to proceed once that decision has been made. These issues notwithstanding, the results of this research must be viewed as encouraging.

Figure 1

COURTYARD WITH OPENINGS ON EACH FLANKING WALL

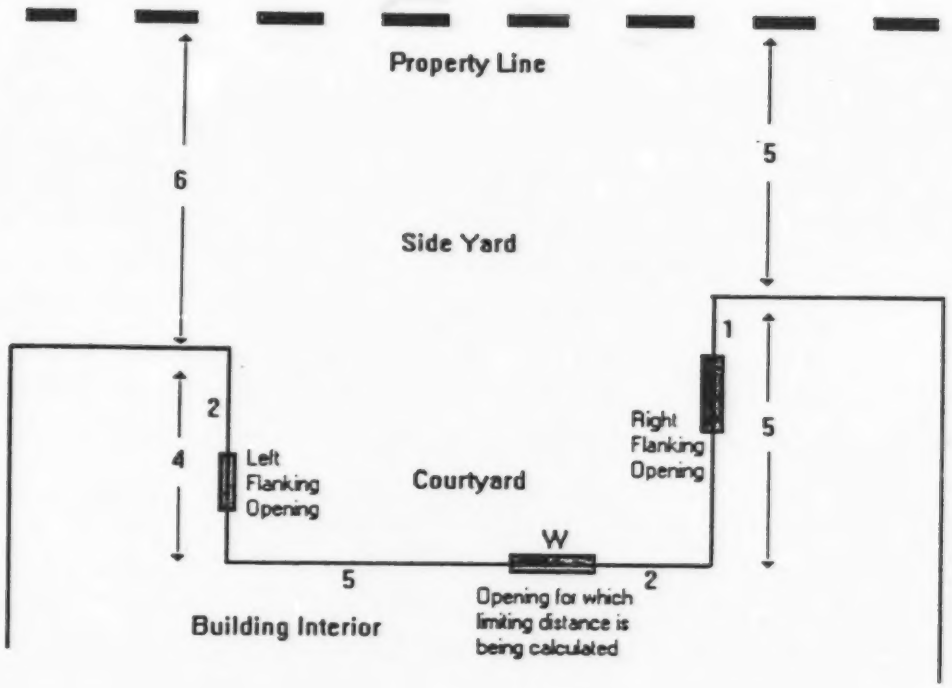


Figure 2

SAMPLE CODE FOR ONE RULE

```
RULE   COURTYD_WLEFT_PROJ_RIGHT_DOMINATES

       IF single_plane = no
       AND on_facing_wall = yes
       AND length_of_flanking_wall_left > 0
       and length_of_flanking_wall_right > 0
       and opening_in_flanking_wall_left = yes
       and opening_in_flanking_wall_right = no
       AND projection_right <> unknown
       AND projection_left >= (projection_right)
       AND flanking_opening_dist_left >= (projection_right)
       THEN limiting_dist = (projection_right);
       ! FPM-14
```

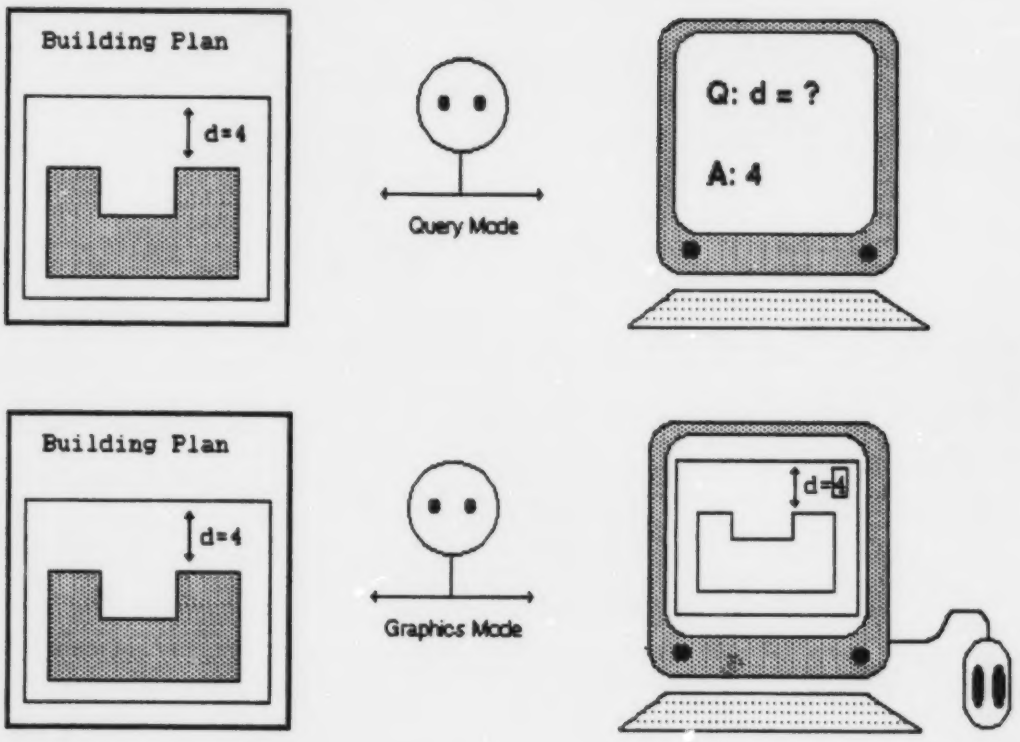
Figure 3

Exerpt of Logical Trace for Consultation

```
Testing STITCH1.kbs
(= yes CNF 0 )
!  appropriate_message
!  !  Testing appropriate_message_S
!  !  !  type_of_analysis
!  !  !  !  (= S CNF 100 )
!  !  (= as_follows CNF 100 )
!  number_of_planes
!  !  Testing number_of_planes
!  !  !  single_plane
!  !  !  !  (= yes CNF 100 )
!  !  (= 1 CNF 100 )
!  min_allowable_dist
!  !  Testing min_allowable_dist_2_ft
!  !  !  building_sprinklered
!  !  !  !  (= no CNF 100 )
!  !  Testing min_allowable_dist_3_or_4_ft
!  !  (= 4 CNF 100 )
!  exposing_building_face
!  !  (= 950 CNF 100)
(= exposing_building_face CNF 100 )
(= 999 CNF 100 )
(= 0 CNF 100 )
```

Figure 4

Query vs. Graphics Mode of User Interface



References

Benfer, Robert A., Brent, Edward E. Jr., and Louanna Furbee. 1991. Expert Systems. Quantitative Applications in the Social Sciences, vol. 77, Sage Publications, Newbury Park.

City of Vancouver. 1991. Evaluation of Prototype Plan Checking Expert System. Internal Memorandum, Department of Permits & Licenses.

Goodall, Alex. 1985. The Guide to Expert Systems. Oxford, UK: Learned Information.

Han, S.Y. and T.John Kim. 1989. Can Expert Systems Help with Planning? Journal of the American Planning Association 55, 3: 296-307.

Heikkila, Eric J. 1990. Modelling Fiscal Impacts Using Expert GIS: Theory and Strategy. Computers, Environment and Urban Systems 14, 1: 25-33.

Hosking, J.G., W.B. Mugridge and M. Buis. 1987. Firecode: A Case Study in the Application of Expert Systems Techniques to a Design Code. Environment and Planning B 14: 267-280.

Kim, T. John, Lyna L. Wiggins and Jeffrey R. Wright (eds.). 1989. Expert Systems: Applications to Urban Planning. New York: Springer-Verlag.

Leary, Michael E. 1989. Expertise and Expert Systems in British Land Use Planning: Analysis and Progress Towards a Prototype. Paper presented to the First International Conference on Expert Systems in Environmental Planning and Engineering. Cambridge, MA.

Ortolano, Leonard and Catherine D. Perman. 1987. A Planner's Introduction to Expert Systems. Journal of the American Planning Association 53, 1: 98-103.

Pedersen, Ken. 1989. Expert Systems Programming: Practical Techniques for Rule-Based Systems. New York: Wiley.

Sergot, M.J. et al. 1986. The British Nationality Act as a Logic Program. Communications of the ACM 29: 270-386.

Silverman, B.G. 1987. Should a Manager Hire an Expert System? Expert Systems for Business. Reading, MA: Addison-Wesley.

Tello, Ernest R. 1988. Mastering AI Tools and Techniques. Indianapolis: Howard W. Sams & Co.

Waterman, Donald. 1986. A Guide to Expert Systems. Reading, MA: Addison-Wesley.

Wiggins, Lyna L. et al. 1989. Expert Systems Technology; in Kim, Wiggins and Wright (eds.), Expert Systems: Applications to Urban Planning. New York: Springer-Verlag.